Amendment to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Currently amended) A machine-implemented method comprising:

receiving, by a first process in a first ~~user~~ virtual memory address space, a shortcut to a physical address associated with a level of a multi-level virtual address translation table;

posting a descriptor, the descriptor comprising a virtual address in the first ~~user~~ virtual memory address space and the shortcut, to ~~an~~ a virtual interface between the first process and a second process, wherein the second process is in a second ~~user~~ virtual memory address space that is different from the first virtual memory address space; and

determining, by the second process, the physical address corresponding to the virtual address associated with the first process based on at least the virtual address and the shortcut.

2. (Previously Presented) The method of claim 1 further comprising transferring data to or from a buffer located at the physical address.

3. (Original) The method of claim 1 further comprising:

generating the shortcut by a third process.

4. (Original) The method of claim 3 wherein generating the shortcut by the third process comprises:

receiving a request to register a virtual buffer, the request including a virtual address corresponding to the start of the virtual buffer;

determining the physical address of one level of the multi-level address translation table associated with the virtual memory space in which the virtual buffer resides; and generating a shortcut based on the physical address of the one level of the multi-level address translation table.

5. (Original) The method of claim 4 wherein generating a shortcut further comprises:

generating the shortcut based on a key unknown to the first process.

6. (Original) The method of claim 4 wherein generating a shortcut further comprises:

generating the shortcut based on a function unknown to the first process.

7. (Original) The method of claim 1 further comprising:

retrieving a key by the second process; and

applying the key to the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

8. (Previously Presented) The method of claim 1 further comprising determining if the physical address is associated with the virtual address.

9. (Original) The method of claim 1 further comprising determining if the virtual page containing the virtual address is pinned into physical memory.

10. (Canceled).

11. (Currently amended) The method of claim 1 further comprising determining if the first process is authorized to access the virtual address, based in part, on a protection table maintained by the second process.

12. (Currently amended) The method of claim 1 further comprising determining if descriptors posted to the interface between the first process and second process are authorized to access the virtual address, based in part, on a protection table maintained by the second process.

13. (Original) The method of claim 1 further comprising:

receiving, by a first process, a plurality of shortcuts, each shortcut to a physical address associated with a level of a multi-level virtual address translation table.

14. (Original) The method of claim 4 wherein generating a shortcut comprises:

applying a function, F, to the physical address of the one level and a key.

15. (Original) The method of claim 14 wherein the key is associated with the interface between the first and second process.

16. (Original) The method of claim 14 wherein the key is associated with the first process.

17. (Currently amended) A machine-implemented method comprising:

generating, by a first ~~user~~ process in a first ~~user~~ virtual memory address space, a request to register a virtual buffer and posting a descriptor to a virtual interface, wherein the virtual buffer is in the first ~~user~~ virtual memory address space and is mapped to physical memory by a multi-level virtual address translation table associated with the first ~~user~~ process;

identifying a block of memory that includes the physical address corresponding to the start of the virtual buffer;

generating, by a second process, one or more shortcuts that map the block of memory that includes the physical address corresponding to the start of the virtual buffer; and transmitting a request to a third ~~user~~ process in a second ~~user~~ virtual memory address space to perform an input or output operation on the virtual buffer, wherein the request includes the shortcut and a virtual address associated with the virtual buffer.

18. (Previously Presented) The method of claim 17 wherein generating a shortcut further comprises:

generating the one or more shortcuts based on a key that is unknown to the first process.

19. (Previously Presented) The method of claim 17 wherein generating a shortcut further comprises:

generating the shortcut based on a function that is unknown to the first process.

20. (Currently amended) The method of claim 17 further comprising determining the physical address of the virtual address based on the virtual address and the shortcut~~, wherein the request includes the shortcut and a virtual address associated with the virtual buffer~~.

21. (Previously Presented) The method of claim 20 further comprising:

determining if the physical address is associated with the virtual address; and

if the physical address is associated with the virtual address, then enabling the input or output operation on at least part of the virtual buffer.

22. (Previously Presented) The method of claim 20 further comprising:

determining if physical pages associated with the physical address are pinned into physical memory; and

if the associated virtual pages are pinned into physical memory, then enabling the input or output operation.

23. (Currently amended) The method of claim 20 further comprising:

determining if the ~~third user~~ first process is authorized to access the associated virtual buffer, based in part, on a protection table maintained by the third process; and

if the ~~third user~~ first process is authorized to access the associated virtual buffer, then enabling the input or output operation on at least part of the virtual buffer.

24. (Currently amended) The method of claim 20 further comprising:

determining if requests posted to the interface between the first ~~user~~ process and the third ~~user~~ process are authorized to access the associated virtual buffer, based in part, on a protection table maintained by the third process; and

if requests to the interface are authorized to access the associated virtual buffer, then enabling the input or output operation on at least part of the virtual buffer.

25. (Canceled).

26. (Currently amended) A system comprising:

a first processor configured to:

execute instructions of a first process which causes the first processor to produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table, and

execute instructions of a second ~~user~~ process in a first ~~user~~ virtual memory address space which causes the first processor to post a descriptor comprising a virtual address and the shortcut to ~~an~~ a virtual interface; and

a second processor configured to execute instructions of a third ~~user~~ process in a second

~~user~~ virtual memory address space which cause the second processor to:

read the descriptor posted on the virtual interface, and

determine a physical address of the virtual address based on at least the virtual address and the shortcut,

wherein the virtual interface is between the second process and the third process and the first virtual memory address space is different from the second virtual memory address space.

27. (Original) The system of claim 26 wherein the instructions of the first process cause the first processor to encrypt the shortcut with a key.

28. (Currently amended) The system of claim 27 wherein the instructions of the third ~~user~~ process cause the second processor to:

retrieve the key; and

apply the key to the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

29. (Currently amended) The system of claim 28 wherein the instructions of the third ~~user~~ process cause the second processor to determine if the physical address is associated with the second ~~user~~ process.

30. (Currently amended) The system of claim 28 wherein the instructions of the third ~~user~~ process cause the second processor to determine if physical pages associated with the physical address are pinned into physical memory.

31. (Currently amended) The system of claim 28 wherein the instructions of the third ~~user~~ process cause the second processor to determine if the second ~~user~~ process is authorized to access the virtual buffer, based in part, on a protection table maintained by the third process.

32. (Currently amended) The system of claim 27 wherein the instructions of the third user process cause the second processor to determine if requests posted to the interface between the second ~~user~~ process and the third ~~user~~ process are authorized to access the virtual buffer, based in part, on a protection table maintained by the third process.

33. (Currently amended) A computer program product ~~residing on a computer readable medium having instructions stored thereon that, when executed by the processor, cause that processor to~~ tangibly embodied in a machine-readable storage device, the computer program product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

~~produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table; and~~

~~write a descriptor comprising a virtual address and the shortcut to an interface between a first user process in a first user virtual memory address space and a second user process in a second user virtual memory address space~~

receiving, by a first process in a first virtual memory address space, a shortcut to a physical address associated with a level of a multi-level virtual address translation table;

posting a descriptor, the descriptor comprising a virtual address in the first virtual memory address space and the shortcut, to a virtual interface between the first process and a second process, wherein the second process is in a second virtual memory address space that is different from the first virtual memory address space; and determining, by the second process, the physical address corresponding to the virtual address associated with the first process based on at least the virtual address and the shortcut.

34. (Currently amended) The computer program product of claim 33 ~~having instructions that further cause the processor to~~, wherein the operations further comprise:

~~encrypt~~ encrypting the shortcut with a key.

35. (Currently amended) The computer program product of claim 33 ~~having instructions that further cause the processor to~~, wherein the operations further comprise:

~~encrypt~~ encrypting the shortcut with a function.

36. (Currently amended) A computer program product ~~residing on a computer readable medium having instructions stored thereon that, when executed by a processor performing operations in a first user virtual memory address space, cause that processor to~~ tangibly embodied in a machine-readable storage device, the computer program product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

~~read a message posted on an interface by a first user process in a different user virtual memory address space, the message including a shortcut to a physical address associated with a level of a multi-level virtual address translation table;~~

~~determine a physical address of a virtual address in the different user virtual memory address space based on at least the virtual address and the shortcut; and~~

~~transmit a message over a network based on contents of the physical address~~

generating, by a first process in a first virtual memory address space, a request to register a virtual buffer and posting a descriptor to a virtual interface, wherein the virtual buffer is in the first virtual memory address space and is mapped to physical memory by a multi-level virtual address translation table associated with the first process;

identifying a block of memory that includes the physical address corresponding to the start of the virtual buffer;

generating, by a second process, one or more shortcuts that map the block of memory that includes the physical address corresponding to the start of the virtual buffer; and

transmitting a request to a third process in a second virtual memory address space to perform an input or output operation on the virtual buffer, wherein the request includes the shortcut and a virtual address associated with the virtual buffer.

37. (Currently amended) The computer program product of claim 36 ~~having instructions that further cause the processor to~~, wherein the operations further comprise:

~~retrieve~~ retrieving a key; and

~~apply~~ applying the key to the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

38. (Currently amended) The computer program product of claim 36 ~~having instructions that further cause the processor to~~, wherein the operations further comprise: ~~determine~~ determining, by the third process, if the physical address is associated with the virtual address.

39. (Currently amended) The computer program product of claim 36 ~~having instructions that further cause the processor to~~, wherein the operations further comprise:

~~determine~~ determining if virtual pages referenced by the ~~message~~ request to the third process are pinned in physical memory.

40. (Currently amended) The computer program product of claim 36 ~~having instructions that further cause the processor to~~, wherein the operations further comprise:

~~determine~~ determining if the first process is authorized to access a virtual buffer referenced by the ~~message~~ request to the third process, based in part, on a protection table maintained by the third process.

41. (Currently amended) The computer program product of claim 36 ~~having instructions that further cause the processor to~~, wherein the operations further comprise:

~~determine~~ determining if ~~messages~~ the descriptor posted on the virtual interface ~~are~~ is authorized to access the virtual buffer, based in part, on a protection table maintained by the third process.

42. (Currently amended) A system comprising:

a client computer; and

a server in communication with the client computer using a network, the server comprising:

a first processor configured to produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table and write a descriptor comprising a virtual address in a first ~~user~~ virtual memory address space and the shortcut to ~~an~~ a virtual interface; and

a second processor configured to perform operations in a second ~~user~~ virtual memory address space, the operations including reading the descriptor posted on the virtual interface, determining a physical address of the virtual address based on at least the virtual address and the shortcut, and transferring data located at the physical address to the client computer using the network.

43. (Previously presented) The system of claim 42 wherein the first processor is configured to encrypt the shortcut with a key.

44. (Previously presented) The system of claim 43 wherein second processor is configured to decrypt the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

45. (Canceled).

46. (Currently amended) A system comprising:

a storage device; and

a server in communication with the storage device over a network, the server comprising:

a first processor configured to produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table and write a descriptor comprising a virtual address in a first ~~user~~ virtual memory address space and the shortcut to ~~an~~ a virtual interface, and

a second processor configured to perform operations in a second ~~user~~ virtual memory address space, the operations including reading the descriptor posted on the virtual interface, determining a physical address of the virtual address based on at least the virtual address and the shortcut, and transferring data located at the physical address to the storage device using the network.

47. (Previously presented) The system of claim 46 wherein the first processor is configured to encrypt the shortcut with a key.

48. (Previously presented) The system of claim 47 wherein second processor is configured to decrypt the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

49. (Canceled).